

TRANSMITTAL FORM

Attorney Docket No.
SVL920010042US1
2168PAF
epwIn re the application **Tammie DANG**Confirmation No: **2261**Serial No: **09/944,939**Group Art Unit: **2172**Filed: **August 31, 2001**Examiner: **To, Baoquoc N.**For: **Method and System for Dynamically Changing Cursor Attributes in an Embedded SQL Application**

ENCLOSURES (check all that apply)

<input type="checkbox"/>	Amendment/Reply	<input type="checkbox"/>	Assignment and Recordation Cover Sheet	<input type="checkbox"/>	After Allowance Communication to Group
<input type="checkbox"/>	After Final	<input type="checkbox"/>	Part B-Issue Fee Transmittal	<input type="checkbox"/>	Notice of Appeal
<input type="checkbox"/>	Information disclosure statement	<input type="checkbox"/>	Letter to Draftsman	<input checked="" type="checkbox"/>	Appeal Brief (in triplicate)
<input type="checkbox"/>	Form 1449	<input type="checkbox"/>	Drawings	<input type="checkbox"/>	Status Letter
<input type="checkbox"/>	(X) Copies of References	<input type="checkbox"/>	Petition	<input checked="" type="checkbox"/>	Postcard
<input type="checkbox"/>	Extension of Time Request *	<input type="checkbox"/>	Fee Address Indication Form	<input type="checkbox"/>	Other Enclosure(s) (please identify below):
<input type="checkbox"/>	Express Abandonment	<input type="checkbox"/>	Terminal Disclaimer		
<input type="checkbox"/>	Certified Copy of Priority Doc	<input type="checkbox"/>	Power of Attorney and Revocation of Prior Powers		
<input type="checkbox"/>	Response to Incomplete Appln	<input type="checkbox"/>	Change of Correspondence Address		
<input type="checkbox"/>	Response to Missing Parts	*Extension of Term: Pursuant to 37 CFR 1.136, Applicant petitions the Commissioner to extend the time for response for xxxxx month(s), from to .			
<input type="checkbox"/>	Executed Declaration by Inventor(s)				

CLAIMS

FOR	Claims Remaining After Amendment	Highest # of Claims Previously Paid For	Extra Claims	RATE	FEE
Total Claims	0	0	0	\$18.00	\$ 0.00
Independent Claims	0	0	0	\$88.00	\$ 0.00
Total Fees					\$ 0.00

METHOD OF PAYMENT

<input type="checkbox"/>	Check no. _____ in the amount of \$ _____ is enclosed for payment of fees.
<input checked="" type="checkbox"/>	Charge \$ <u>340.00</u> to Deposit Account No. <u>09-0460</u> (IBM Corporation) for payment of Appeal Brief filing fee.
<input checked="" type="checkbox"/>	Charge any additional fees or credit any overpayment to Deposit Account No. <u>09-0460</u> (IBM Corporation)

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Attorney Name	Joseph A. Sawyer, Jr., Reg. No. 30,801	10/20/2004 HLE333 00000034 090460 09944939
Signature		01 FC-1402 340.00 DA
Date	October 14, 2004	

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on October 14, 2004	
Type or printed name	Irena Nikolova
Signature	



**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

APPEAL NO:

In Re Application of: Tammie DANG

Confirmation No. 2261

Serial No. 09/944,939

Filed: August 31, 2001

For: METHOD AND SYSTEM FOR DYNAMICALLY CHANGING CURSOR
ATTRIBUTES IN AN EMBEDDED SQL APPLICATION

APPELLANT'S BRIEF

10/20/2004 HLE333 00000034 09944939

01 FC:1402 340.00 DA

Attorney for Appellants
SAWYER LAW GROUP LLP
P.O. Box 51418
Palo Alto, CA 94303

TOPICAL INDEX

I. REAL PARTY IN INTEREST

II. RELATED APPEALS AND INTERFERENCES

III. STATUS OF CLAIMS

IV. STATUS OF AMENDMENTS

V. SUMMARY OF THE INVENTION

VI. ISSUES

VII. GROUPING OF CLAIMS

VIII. ARGUMENTS

- A. Summary of the Applied Rejections
- B. The Cited Prior Art
- C. Claims 1-20 Are Not Unpatentable Under 35 U.S.C. 102(b)
- D. Summary of Arguments

IX. APPENDIX

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In Re Application of:

Date: October 14, 2004

Tammie DANG

Confirmation No. 2261

Serial No. 09/944,939

Group Art Unit: 2172

Filed: August 31, 2001

Examiner: To, Baoquoc N.

For: METHOD AND SYSTEM FOR DYNAMICALLY CHANGING CURSOR
ATTRIBUTES IN AN EMBEDDED SQL APPLICATION

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

APPELLANT'S BRIEF ON APPEAL

Sir:

Appellant herein files an Appeal Brief drafted in accordance with the provisions
of 37 C.F.R. § 1.192(c) as follows:

I. REAL PARTY IN INTEREST

Appellant respectfully submits that the above-captioned application is assigned, in
its entirety to International Business Machines Corporation, Armonk, New York.

II. RELATED APPEALS AND INTERFERENCES

Appellant states that, upon information and belief, he is not aware of any co-pending appeal or interference which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Application Serial No. 09/944,939 (the instant application) as originally filed included claims 1-20. Claims 1-20 are pending. Claims 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, and 20 are on appeal and all applied prospective rejections concerning Claims 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, and 20 are being appealed herein.

IV. STATUS OF AMENDMENT

No amendments were made to the instant application.

V. SUMMARY OF THE INVENTION

The present invention provides aspects for dynamically changing attributes in an embedded-SQL application. In these aspects, an option is provided within a standard SQL statement for specifying one or more attributes of at least a declared cursor. The standard SQL statement is then processed to include the specified one or more attributes in at least the declared cursor. The provision occurs with an ATTRIBUTES option and within a PREPARE SQL statement.

Through the present invention, a standard approach to statement preparation is enhanced in a straightforward manner to achieve greater flexibility for dynamically changing attributes in an embedded-SQL application. With the present invention, only a small number of pre-declared cursors are needed in a driver program. When a user application needs a particular set of attributes, the driver (usually running in the middle tier) can dynamically prepare the SELECT statement and, at the same time, re-prepare these attributes with the cursor. In this manner, the cursors become reusable in the execution environment, and the space required is optimized.

VI. ISSUES

The issue presented is:

(1) whether claims 1-20 are unpatentable under 35 U.S.C. 102(b) over Jasuja et al. (“Jasuja”).

VII. GROUPING OF CLAIMS

Appellant hereby states that claims 1-20 form one group.

VIII. ARGUMENTS

A. Summary of the Applied Rejections

In the final office action dated 3/9/04, the Examiner appears to have not included a specific indication of the rejection, but based on the Examiner’s response to Appellant’s previous arguments, Appellant believes that the Examiner maintained the rejection of claims 1-20 under 35 U.S.C. 102(b) as being anticipated by Jasuja. Further, in response

to Appellant's arguments regarding the rejection, the Examiner respectfully disagrees with the Appellant's position and states:

The examiner respectfully disagrees ... because Jasuja suggests, "pseudo SQL statements and corresponding pseudo PL/SQL statement, which when executed by a client, cause a cursor to be created. Of course, additional statements could be intermixed with the exemplary statements shown in Table 3 (col. 8, lines 54-56). This teaches the cursor is created dynamically wherein which is the better system then the present [sic] wherein the user specifying the attributes for the declared cursor.

Appellant respectfully requests that the Board reverse the Examiner's final rejection of the pending Claims.

B. The Cited Prior Art

Jasuja discloses a method and apparatus for reducing the memory required to store bind variable descriptors in a database system. In the embodiments, bind variable descriptors continue to be maintained in the shared cursor objects, but bind variables are no longer maintained in the instantiation objects. If the bind variables associated with an instantiation object are accurately described by the bind variable descriptors in a shared object to which the instantiation object is linked, then only data indicating the bind variables associated with the instantiation object is maintained in the instantiation object. In contrast, if the bind variable descriptors associated with an instantiation object are not accurately described, only data describing how the attributes of the bind variables associated with the instantiation object differ from those associated with the shared cursor object are maintained in the instantiation object.

C. Claims 1-20 Are Not Unpatentable Under 35 U.S.C. 102(b)

The present invention dynamically changes attributes in an embedded-SQL application with the provision of an option within a standard SQL statement for specifying one or more attributes of at least a declared cursor and the processing of the standard SQL statement to include the one or more attributes in the at least declared cursor. See independent claims 1, 8, and 15. Appellant recites more particularly that the option is provided as an ATTRIBUTES option within a PREPARE SQL statement. See dependent claims 2, 3, 9, 10, 16, and 17. In this manner, attributes of a cursor may be dynamically changed without requiring declaration of a new cursor or the introduction of a new SQL statement. Appellant illustrates this ability in the SQL examples in the specification from page 8, line 18 to page 10, line 9. Appellant fails to see anything in the Jasuja reference that teaches or suggests the ability to provide attributes of a declared cursor in a standard SQL statement and/or the ability to process the standard SQL statement to include the attributes in the declared cursor.

The cited art of Jasuja is concerned with reducing database memory requirements for storing descriptors for bind variables of cursors. Bind variables are described as the input variables or parameterized values of a predicate of a query that, when changed, change the rows returned by the query (see col. 2, lines 2-20). Jasuja teaches that “bind variable descriptors are data describing attributes (e.g., the data types, lengths, precision, scale, etc., but not the values) of the bind variables” (col. 3, lines 14-17). The aim in Jasuja is to change how bind variable descriptors are maintained by only maintaining them in shared cursor objects and not in instantiation objects.

While reducing memory requirements in database systems is a concern in Jasuja and the present invention, the issue addressed by Jasuja regarding a change in how bind variable descriptors are maintained fails to teach, show, or suggest the present invention, which addresses aspects of specifying attributes of a declared cursor to dynamically change the attributes. In fact, Appellant respectfully submits that the Jasuja reference is silent regarding cursor attributes and thus offers no teaching or suggestion as to how attributes of a declared cursor could or would be specified and processed. While Jasuja does make reference to the association of bind variable attributes for a query with a cursor created to execute the query and an instantiation object for the cursor (cited col. 8, lines 42-53), there is nothing from the mere mention of an association of bind variable attributes with a cursor that teaches or suggests the recited dynamic changing of attributes, including, providing an option within a standard SQL statement for specifying one or more attributes of at least a declared cursor and processing the standard SQL statement to include the specified one or more attributes in the at least declared cursor, as recited in independent claims 1, 8, and 15. More particularly, Jasuja wholly fails to teach, show, or suggest an ATTRIBUTES option provided within a PREPARE SQL statement, as recited in dependent claims 2, 3, 9, 10, 16, and 17.

Further, Jasuja's pseudo SQL statements and corresponding pseudo PL/SQL statements (relied upon in the Examiner's response) cause a cursor to be created. The Appellant's recited invention, however, concerns how attributes of a declared cursor are dynamically changed through an option provided in a standard SQL statement. Appellant fails to see how the creation of a cursor teaches or suggests dynamically changing attributes of a cursor that has been declared. Accordingly, without further criticality of

teaching, Appellant respectfully submits that even the pseudo SQL statements or pseudo PL/SQL statements of Table 3 of Jasuja for cursor creation offer no teaching or suggestion of the dynamic changing of attributes of a declared cursor through an option provided in a standard SQL statement, as recited in independent claims 1, 8, and 15.

In the Advisory Action, the "examiner respectfully disagrees with the above argument, because 'dynamic changing of attributes of a declared cursor' is not what is being claimed and as same as dependent claims 2, 3, 9, 10 16 [sic] and 17." Appellant respectfully directs attention to the recited claims. Independent claim 1 recites "A method for dynamically changing attributes in an embedded-SQL application", while independent claim 8 recites "A system for dynamically changing attributes in an embedded-SQL application", and independent claim 15 recites "A computer readable medium containing program instructions for dynamically changing attributes in an embedded-SQL application." Further, each of the claims recites aspects for the specification and inclusion of attributes for and in at least a declared cursor. Appellant fails to see how the 'dynamic changing of attributes of a declared cursor' is not what is being claimed when considering the entirety of each of the recited claims.

Furthermore, while the Examiner states that all dependent claims are rejected under the same reasons as independent claims 1, 8, and 15, Appellant respectfully submits that a blanket statement such as this fails to provide sufficient basis for the anticipatory rejection of these claims. With particular regard, Appellant reiterates that Jasuja wholly fails to teach, show, or suggest an ATTRIBUTES option provided within a PREPARE SQL statement, as recited in dependent claims 2, 3, 9, 10, 16, and 17. As

such, Appellant fails to see how these dependent claims can be rejected under the same reasons as independent claims 1, 8, and 15.

In view of the foregoing, Appellant respectfully submits that claims 1-20 are not taught, shown, or suggested by the cited art.

Accordingly, Appellant respectfully requests withdrawal of the rejection under 35 U.S.C. 102(b) and respectfully requests that the Board reverse the final rejection of Claim 1-20.

D. Summary of Arguments

For all the foregoing reasons, it is respectfully submitted that Claims 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, and 20 (all the Claims presently in the application) are patentable for defining subject matter which would not have been unpatentable under 35 U.S.C. § 102(b). Thus, Appellant respectfully requests that the Board reverse the rejection of all the appealed Claims and find each of these Claims allowable.

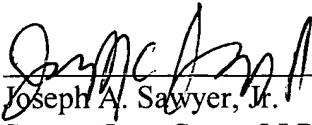
Note: For convenience of detachment without disturbing the integrity of the remainder of pages of this Appeal Brief, Appellant's "APPENDIX" section is contained on separate sheets following the signatory portion of this Appeal Brief.

This Brief is being submitted in triplicate, and authorization for payment of the required Brief fee is contained in the transmittal letter for this Brief. Please charge any fee that may be necessary for the continued pendency of this application to Deposit Account No. 09-0460 (IBM Corporation).

Respectfully submitted,

October 14, 2004

Date



Joseph A. Sawyer, Jr.
Sawyer Law Group LLP
Attorney for Appellant(s)
Reg. No. 30,801
(650) 493-4540

IX. APPENDIX

1. A method for dynamically changing attributes in an embedded-SQL application, the method comprising the steps of:
 - (a) providing an option within a standard SQL statement for specifying one or more attributes of at least a declared cursor; and
 - (b) processing the standard SQL statement to include the specified one or more attributes in at least the declared cursor.
2. The method of claim 1 wherein the option providing step (a) further comprises the steps of (a1) providing an ATTRIBUTES option.
3. The method of claim 2 wherein option providing step (a) further comprises the steps of (a2) providing an option within a PREPARE SQL statement.
4. The method of claim 3 wherein the processing step (b) further comprises the steps of (b1) parsing the ATTRIBUTES option.
5. The method of claim 4 further comprising the step of (c) resolving conflicting and duplicate attributes.

6. The method of claim 5 further comprising the step of (c) placing resolved attributes in a parse tree.

7. The method of claim 1 further comprising the step of (c) utilizing the one or more attributes in a concatenated string for a dynamic cache system of a database server.

8. A system for dynamically changing attributes in an embedded-SQL application, the system comprising:

at least one computer processing device; and

a database management system installed on the at least one computer processing device, the database management system supporting provision of an option within a standard SQL statement for specifying one or more attributes of at least a declared cursor, and processing of the standard SQL statement to include the specified one or more attributes in at least the declared cursor.

9. The system of claim 8 wherein provision of an option further comprises provision of an ATTRIBUTES option.

10. The system of claim 9 wherein provision of an option further comprises provision of an option within a PREPARE SQL statement.

11. The system of claim 10 wherein processing further comprises parsing the ATTRIBUTES option.

12. The system of claim 11 wherein the database management system further supports resolution of conflicting and duplicate attributes.

13. The system of claim 12 wherein the database management system further supports placement of resolved attributes in a parse tree.

14. The system of claim 8 wherein the database management system further supports utilization of the one or more attributes in a concatenated string for a dynamic cache system of a database server.

15. A computer readable medium containing program instructions for dynamically changing attributes in an embedded-SQL application, the program instructions comprising:
providing an option within a standard SQL statement for specifying one or more attributes of at least a declared cursor; and

processing the standard SQL statement to include the specified one or more attributes in at least the declared cursor.

16. The program instructions of claim 15 wherein providing an option further comprises providing an ATTRIBUTES option.

17. The program instructions of claim 16 wherein providing an option further comprises providing an option within a PREPARE SQL statement.

18. The program instructions of claim 17 wherein processing further comprises parsing the ATTRIBUTES option.

19. The program instructions of claim 18 further comprising resolving conflicting and duplicate attributes.

20. The program instructions of claim 19 further comprising placing resolved attributes in a parse tree.